

CLAIMS

We claim:

1 1. A method for identifying erroneous transactions that occur during processor
2 architecture verification testing, the method comprising:
3 monitoring an interface;
4 determining information related to termination of a test case; and
5 after the test case has terminated, identifying an incomplete transaction that should
6 have completed prior to termination of the test case.

1 2. The method of claim 1, wherein monitoring an interface comprises monitoring
2 a point-to-point (P2P) link network.

1 3. The method of claim 1, wherein monitoring an interface comprises monitoring
2 a point-to-point (P2P) link network of a register transfer language (RTL) simulator.

1 4. The method of claim 1, wherein determining information related to
2 termination of a test case comprises detecting a break signal asserted on the interface.

1 5. The method of claim 1, wherein determining information related to
2 termination of a test case comprises receiving an indication that a test model has stopped
3 processing.

1 6. The method of claim 1, wherein identifying an incomplete transaction
2 comprises consulting a pending transactions list.

1 7. The method of claim 6, wherein identifying an incomplete transaction further
2 comprises filtering out pending transactions of a type that does not likely indicate an error.

1 8. The method of claim 7, wherein filtering out pending transactions comprises
2 disregarding at least one of transactions having a start time on or after the time at which a
3 break signal was asserted and transactions that occur as a result of a break command being
4 issued.

1 9. The method of claim 7, wherein identifying an incomplete transaction further
2 comprises flagging all pending transactions other than those that were filtered out.

1 10. The method of claim 9, wherein flagging all other transactions comprises
2 alerting a user as to the existence of the pending transactions and providing information to the
3 user that can be used to determine the reason why those transactions did not complete.

1 11. The method of claim 10, wherein providing information comprises at least one
2 of providing all completed packets associated with the transactions and providing a summary
3 of each transaction that describes all processing associated with each transaction.

1 12. A system for identifying erroneous transactions, the system comprising:
2 means for monitoring all ports of an interface;
3 means for determining information related to termination of a test case; and
4 means for identifying an incomplete transaction that should have completed prior to
5 termination of the test case.

1 13. The system of claim 12, wherein the means for monitoring comprise means
2 for monitoring a point-to-point (P2P) link network of a register transfer language (RTL)
3 simulator.

1 14. The system of claim 12, wherein the means for determining information
2 comprise means for detecting break signals asserted on the interface.

1 15. The system of claim 12, wherein the means for determining information
2 comprise means for receiving an indication that a test model has stopped processing.

1 16. The system of claim 12, wherein the means for identifying an incomplete
2 transaction comprise a pending transactions list.

1 17. The system of claim 16, wherein the means for identifying an incomplete
2 transaction comprise means for disregarding at least one of transactions having a start time on
3 or after the time at which a break signal was asserted and transactions that occur as a result of
4 a break command being issued.

1 18. The system of claim 12, wherein the means for identifying an incomplete
2 transaction comprise means for flagging all pending transactions that are determined to be
3 erroneous.

1 19. The system of claim 18, wherein the means for flagging comprise means for at
2 least one of providing all completed packets associated with the transactions and providing a
3 summary of each transaction that describes all processing associated with each transaction.

1 20. A virtual bus interface (VBI) stored on a computer-readable medium, the VBI
2 comprising:

- 3 logic configured to monitor a point-to-point (P2P) interface;
- 4 logic configured to determine a time at which a break signal was asserted;
- 5 logic configured to identify transactions that are pending after the break signal was
- 6 asserted; and
- 7 logic configured to determine which of the pending transactions are erroneous.

1 21. The VBI of claim 20, wherein the logic configured to identify transactions
2 comprises logic configured to consult a pending transactions list.

1 22. The VBI of claim 20, wherein the logic configured to determine comprises
2 logic configured to filter out pending transactions of a type that does not likely indicate an
3 error.

1 23. The VBI of claim 22, wherein the logic configured to filter out pending
2 transactions comprises logic configured to disregard at least one of transactions having a start
3 time on or after the time at which the break signal was asserted and transactions that occur as
4 a result of a break command being issued.

1 24. The VBI of claim 20, further comprising logic configured to flag erroneous
2 transactions.

1 25. The VBI of claim 24, wherein the logic configured to flag comprises logic
2 configured to alert a user as to the existence of the erroneous transactions and logic
3 configured to provide information to the user that can be used to determine the reason why
4 those transactions did not complete.

1 26. A processor architecture verification system, comprising:
2 a register transfer language (RTL) simulator that simulates operation of a processor
3 and generates a first output in a first format, the RTL simulator including an interface;
4 a golden simulator that simulates operation of the processor and generates a second
5 output in a second format;
6 a translator that translates at least one of the outputs for comparison with the other
7 output, the translator including a virtual bus interface (VBI) that comprises logic configured
8 to monitor a point-to-point (P2P) interface, logic configured to determine a time at which a
9 break signal was asserted, logic configured to identify transactions that are pending after a
10 break command was issued, and logic configured to determine which of the pending
11 transactions are erroneous; and
12 a comparator that compares the first and second outputs after translation of the at least
13 one output.

1 27. The system of claim 26, wherein the logic configured to identify transactions
2 comprises logic configured to consult a pending transactions list.

1 28. The system of claim 26, wherein the logic configured to determine comprises
2 logic configured to filter out pending transactions of a type that does not likely indicate an
3 error.

1 29. The system of claim 26, wherein the logic configured to filter out pending
2 transactions comprises logic configured to disregard at least one of transactions having a start
3 time on or after the time at which the break signal was asserted and transactions that occur as
4 a result of a break command being issued.